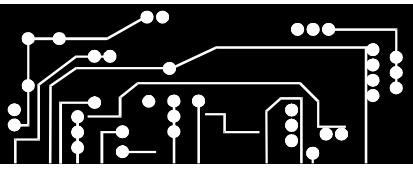

IBM 6xx/7xx Processor Family Frequently Asked Questions (FAQ)

Microcontroller Applications
IBM Microelectronics
Research Triangle Park, NC
ppcsupp@us.ibm.com,
Version: 1.0

September 9, 1998

Abstract - This document addresses topics and questions frequently asked about the PowerPC 603, 603e, 603ev, 740 and 750 processors. This information highlights and compliments the information found in the PPC603e and PPC 740/750 User's Manuals and the associated hardware specifications.

1. What is an Em603e? (9/8/98)

The IBM Em603e is actually a family of processors; EM603e-100, EM603e-166, and EM603e-200. The higher speed processors are implemented in a faster technology.

Each Em603e processor is identical to the 603e processor of the same speed, with the exception that the floating point unit (FPU) and floating point register files are not supported. Table 1 outlines the different Em603e family members:

Table 1: EM603e Family Members

General	603e 100 MHz	603e 166 MHz	603e 200 MHz
Technology	0.5 um CMOS PID6	0.35 um CMOS PID7v	0.35 um CMOS PID7t
Die Size	98 sq mm	79 sq mm	79 sq mm
Number of transistors	2.6 million	2.6 million	2.6 million
Tj (junction temp)	0 - 105 deg C	0 - 105 deg C	0 - 105 deg C
Power Dissipation	3.2/4.0 W @ 100 MHz	3.0/4.0 W @ 166 MHz	4.0/5.0 W @ 200 MHz
Signals	165	165	165
Package	255 CBGA 240 PFP w/MHS	255 CBGA	255 CBGA
Voltages (+/- 5%)	3.3 V Core 3.3 V I/O	2.5 V Core 3.3 V I/O	2.5 V Core 3.3 V I/O
PVR number	0x0006	0x0007	0x0007
Part Number	IBM25EMPPC 603eBG100F, IBM25EMPPC 603ePG100F	IBM25EMPPC 603eBC166F	IBM25EMPPC 603e2BB200F

2. What are the limitations of the Em603e? (9/8/98)

Proper operation of the floating point arithmetic unit is not guaranteed for the Em603e parts. Software should either be modified so that all floating point instructions are removed, or the floating point library software should be used to emulate the floating point instructions. The default state of MSR[FP] = 0 must

not be changed; this default setting will cause a floating point exception to be generated when any floating point load, store, or arithmetic instruction is executed. The exception handler can then call the appropriate library routine to emulate the floating point instruction.

3. What is the difference between the 740 and 750 processors? (9/8/98)

The 740 processor is just a 255 pin version of the 360 pin 750. The 750 has built-in support (cache tags) for a direct attached L2 cache; the pin difference between the two parts is to accommodate the L2 interface. There is a new, faster version of the 740/750 processor that has just been announced. In addition to being faster, the new parts have a different (lower) core voltage. Table 2 outlines the different 740/750 processors:

Table 2: 740/750 Processors

	PID 8t	**PID 8p (new)**
Processor Speed	200/233/266 MHz	300/333/366/400 MHz
Technology	0.25 um/0.18 Leff CMOS	0.22 um/0.12 Leff CMOS
Die Size	67 sq. mm	40 sq. mm
Power Supply (voltages)	2.5 V to 2.7 V core 3.3 V +/- 5% I/O	2.0 V to 2.1 V core (see note) 3.3 V +/- 5% I/O
Power Dissipation (typical)	5.7W @ 266 MHz (740) 5.9W @ 266 MHz (750)	3.7W @ 400 MHz (740) 4.1 W @ 400 MHz (750)
Temperature Range	0 to 105 deg C	0 to 105 deg C (see note)
Packaging	255-pin CBGA (740) 360-pin CBGA (750)	255-pin CBGA (740) 360-pin CBGA (750)

Note: Application conditions exist for higher speed parts; refer to product data sheets for specific application conditions; get data sheets at:

http://chips.ibm.com/techlib/products/embedded/datasheets.html

4. How is 32-bit data mode enabled, and is it supported on the 740/750 processors? (9/9/98)

The bus size is determined at power on, as soon as HRESET# goes inactive and the part comes out of reset. At that time, the processor will sample the state of the TLBSYNC# pin; if TLBSYNC# is asserted at the negation of HRESET#, the processor begins operation in 32-bit data bus mode. See Section 8.6.1 of the *603e User's Manual* for more details.

32-bit data bus mode is not supported today for 740/750 processors, but is presently being verified and support is planned for the near future. Customers may contact the embedded PowerPC support center at ppcsupp@us.ibm.com for specific information.

5. Can DBDIS# be used to halt data bus tenure on the 740/750? (9/8/98)

The DBDIS# signal is just a synchronous output enable/disable control for the data bus pins. It does not affect the data bus flow at all. Processor internal bus interface logic will still look at all of the bus control signals independent of the state of DBDIS#. **Use of the DBDIS# signal is not recommended for new designs.**

6. What is the difference between HRESET# and SRESET# ? (9/8/98)

6.1 HRESET#

HRESET#, the hard reset signal, is considered to be a non recoverable, nonmaskable exception. HRESET# must be used at power-on in conjunction with the TRST# signal to properly reset the processor. HRESET# can also occur at any time and may be asserted asynchronously to the processor input (SYSCLK) clock. It must be held asserted for a minimum of 255 clock cycles after the PLL lock time (100 us) has been met.

A hard reset will initialize latches, zero out some registers, and leave others in an indeterminate state. HID0, HID1, SRR0 and SRR1 registers will be all 0s, and the MSR will be set to 0x00000040 (IP set so that hard reset exceptions will always vector to address 0xfff00100). The BATs, segment registers, and TLBs will all be left in indeterminate states. The processor stays in this state while HRESET# is asserted. When HRESET# goes inactive, the processor begins operation by issuing an instruction fetch from address 0xfff00100. Because caching is inhibited at startup, this will be a single beat read operation.

6.2 SRESET#

SRESET#, the soft reset signal, is considered to be a non recoverable, nonmaskable exception. SRESET# can occur at any time and may be asserted asynchronously to the processor's input (SYSCLK) clock. The SRESET# input is negative edge sensitive and may be negated two cycles after assertion.

The assertion of SRESET# generates a system reset exception, and the processor will attempt to reach a recoverable state before taking the exception. The exception taking process modifies the MSR, SRR0, and SRR1, as described in the *Programming Environments Manual* and Chapter 4 of the *603e User's Manual* and *740/750 User's Manual*. SRESET# does not directly affect the states of off chip (output) signals. The MSR[IP] bit is not cleared by a SRESET# exception, so the exception handling for soft resets will vector to either high (0xfff00100) or low (0x00000100) memory, depending on system software design.

7. How should HRESET# and TRST# be wired to allow RISCWatch to be used? (9/8/98)

TRST# is the reset input to the processor JTAG interface. The JTAG logic must be reset during power on (hard reset) to ensure that the JTAG logic does not interfere with the operation of the processor.

Basically, the system must assert TRST# whenever it asserts HRESET#. When using the RISCWatch debugger, there is a TRST# output and a HRESET# output from the RISCWatch JTAG probe that needs to be taken into account. These signals are typically handled as shown in Figure 1.

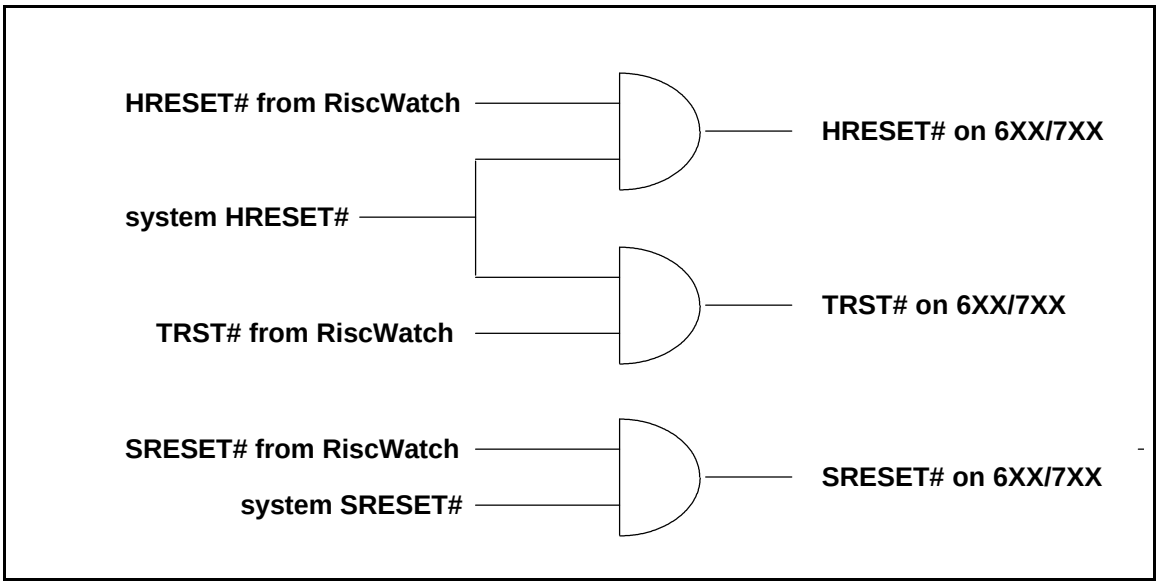

Figure 1. RISCWatch JTAG to HRESET#, TRST#, and SRESET# Signal Connections

Figure 1 shows how to wire the HRESET#, TRST#, and SRESET# signals from the RiscWatch JTAG connector with the HRESET# and SRESET# signals generated by system logic. This allows the CPU to be reset by either the hardware (normal operation) or the RiscWatch (for debug). The addition of weak (10K ohm) pullups on the lines from the RISCWatch JTAG connector will prevent spurious resets when the RiscWatch is not attached.

8. What registers should be initialized before use? (9/8/98)

Table 4-9 in the *740/750 User's Manual* and Table 4-8 in the *603e User's Manual* list the register values caused by a HRESET# or a power-on condition. **All GPRs, FPRs, SRs, TLBs, and BATs come up in undefined states and must be initialized prior to use.**

Unused BATs must always be initialized to 0x00000000.

FPR initialization prior to use is strongly recommended. FPRs must be initialized prior to execution of stfd instructions to prevent potential processor hangs due to the error conditions caused by execution with undefined operands.

9. How is page mode translation disabled? (9/8/98)

There is no trivial way to disable page based virtual memory since:

- Segment registers do not have a valid bit.
- There is no single bit anywhere (like in HID0) where it can be disabled.

Block (BAT) and Page translation occur simultaneously, and a successful Block translation always takes precedence. The minimum page table size is 64K bytes (*PowerPC Programming Environments Manual*,

Chapter 7), so always reserve 64K (minimum) for the page table. To setup a minimal size page table with a minimal amount of effort, this block must be 64K byte aligned and filled with 8K invalid Page Table Entries (PTEs). Zeroed memory suits this purpose, since the Valid bit in the lower-addressed word will always be zero.

Once this 64K area is cleared, move the real (non-translated) address of this area into SDR1. You can do this real address copy ONLY if your block is aligned on a 64K byte boundary. This works because the HTABMASK portion of SDR1 is supposed to be all zeroes for a minimally sized page table, and since the page table is 64K byte aligned, the low 16 bits will already be zero.

There is no requirement to clear the Segment registers. Because of the mechanism by which a Page Table Entry group is selected (see Section 7.6.1.4 in the *PowerPC Programming Environments Manual* for details), it might minimize the L1 D-cache footprint to use the same value for each segment register. In any case, having them all the same will not adversely impact the cache footprint.

To recap, while translation is OFF, do the following three steps:

1. Zero the page table (64K bytes aligned on a 64K boundary)
2. Set SDR1 to point to the page table
3. Invalidate the TLBs.

Optionally, clear the segment registers.

It does not matter where the page table setup occurs relative to the BAT setup, as long as translation is turned off during both setup operations.

10. Where can I find models for these processors? (9/8/98)

Both IBIS and BSDL can be found on the web at:

http://www.chips.ibm.com/techlib/products/embedded/models.html

© International Business Machines Corporation, 1998

All Rights Reserved

* Indicates a trademark or registered trademark of the International Business Machines Corporation.

** All other products and company names are trademarks or registered trademarks of their respective holders.

IBM and IBM logo are registered trademarks of the International Business Machines Corporation.

IBM will continue to enhance products and services as new technologies emerge. Therefore, IBM reserves the right to make changes to its products, other product information, and this publication without prior notice. Please contact your local IBM Microelectronics representative on specific standard configurations and options.

IBM assumes no responsibility or liability for any use of the information contained herein. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of IBM or third parties. NO WARRANTIES OF ANY KIND, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE ARE OFFERED IN THIS DOCUMENT.